

**REMARKS**

Applicant was required to divide FIGS. 9 and 20 into three sheets, FIGS. 11, 12 and 14 into four sheets and FIGS. 13 and 15 into five sheets. The specification has been amended to reflect such change.

The Assistant Commissioner is hereby authorized to charge payment of any additional fees associated with this communication or credit any overpayment to Deposit Account No. 500845.

Respectfully submitted,

Date

7/9/01



Richard M. Sharkansky  
Attorney for Applicant(s)  
Registration No. 25,800  
Daly, Crowley & Mofford, LLP  
275 Turnpike Street, Suite 101  
Canton, MA 02021-2310  
Telephone (781) 401-9988 x23  
Facsimile (781) 401-9966

RECEIVED

JUL 17 2001

GROUP 2100

RECEIVED

JUL 17 2001

Group 2100

Q:\emc2\EMC2-087PUS\emc2-087pus-pm amn for formal drs.doc

Attachment: 21 sheets of Specification changes with markings showing changes made.



**VERSION OF SPECIFICATION PARAGRAPHS WITH MARKINGS SHOWING**  
**CHANGES MADE**

**Change to the Paragraph beginning on Page 5 at line 25.**

FIG. 6 shows the relationship between FIGS. 6A and 6B which when taken together is a block diagram showing the connections between front-end and back-end directors to one of a pair of message network boards used in the system interface of the data storage system of FIG. 2;

**Change to the Paragraphs beginning on Page 6 at line 7.**

FIGS. 9A, 9B and 9C isare a more detailed block diagram of the exemplary cache memory board of FIG. 8A;

FIG. 10 is a block diagram of a crossbar switch used in the memory board of FIGS. 9A, 9B and 9C;

FIGS. 11A, 11B, 11C and 11D isare a block diagram of an upper port interface section used in the crossbar switch of FIG. 10;

FIGS. 12A, 12B, 12C and 12D isare a block diagram of a lower port interface section used in the crossbar switch of FIG. 10;

FIGS. 13A, 13B, 13C, 13D and 13E isare a block diagram of a pair of logic sections used in the memory board of FIGS. 9A, 9B and 9C;

FIGS. 14A, 14B, 14C and 14D isare a block diagram of a pair of port controllers used in the pair of logic sections of FIGS. 13A, 13B, 13C, 13D and 13E;

FIGS. 15A, 15B, 15C, 15D and 15E isare a block diagram of a pair of arbitration logics used in the pair of logic sections of FIGS. 13A, 13B, 13C, 13D and 13E and of a watchdog section used for such pair of logic sections;

FIG. 16 is a diagram showing words that make up exemplary information cycle used in the memory board of FIGS. 9A, 9B and 9C;

FIG. 17 is a Truth Table for a majority gate used in the memory board of FIGS. 9A, 9B and 9C;

FIG. 18 is a block diagram shown interconnections between one of the arbitration units used in one of the pair of port controllers of FIGS. 13A, 13B, 13C, 13D and 13E and a filter used in the arbitration unit of the other one of such pair of controllers of FIGS. 13A, 13B, 13C, 13D and 13E;

FIG. 19 is a timing diagram of signals in arbitration units of FIG. 18 used of one of the pair of port controllers of FIGS. 14A, 14B, 14C and 14D and a filter used in the arbitration unit used in the other one of such pair of controllers of FIGS. 14A, 14B, 14C and 14D; and

FIGS. 20A, 20B and 20C ~~is~~are a more detailed block diagram of arbitrations used in the arbitration logics of FIGS. 15A, 15B, 15C, 15D and 15E.

**Change to the Paragraph beginning on Page 20 at line 6.**

Referring again to FIG. 8, the system includes a plurality of, here eight, memory boards. As described above in connection with FIG. 8A, each one of the memory boards includes four memory array regions  $R_1$ - $R_4$ . Referring now to FIGS. 9A, 9B and 9C, an exemplary one of the cache memory boards in the cache memory 220 (FIG. 8), here cache memory board 220<sub>1</sub>, is shown in more detail to include, here, the four logic networks 221<sub>1B</sub>, 221<sub>2B</sub>, 221<sub>1A</sub>, and 221<sub>2A</sub> and, here eight interface, or memory region control, sections, here logic sections 5010<sub>1</sub>-5010<sub>8</sub>, arranged as shown.

**Change to the Paragraphs beginning on Page 21 at line 30.**

More particularly, each one of the crossbar switches 5004<sub>1</sub>-5004<sub>4</sub> has, here, four lower ports 5008<sub>1</sub>-5008<sub>4</sub> and four upper ports 5006<sub>1</sub>-5006<sub>4</sub>. Each one of the four upper ports 5006<sub>1</sub>-5006<sub>4</sub>, is, as noted above, coupled to a corresponding one of the four sets  $S_1$ - $S_4$ , respectively, of four of the S/P converters. As noted above, the cache memory board 220<sub>1</sub> also includes eight logic sections coupled 5010<sub>1</sub> - 5010<sub>8</sub> (to be described in detail in connection with FIGS. 13A, 13B, 13C, 13D and 13E) as well as the four memory array regions  $R_1$ - $R_4$ . An exemplary one of the memory array regions  $R_1$ - $R_4$  is described in connection with FIG. 6 of U. S. Patent No. 5,943,287. As described in such U. S. Patent, each one of the memory array regions includes a

pair of redundant control ports  $P_A$ ,  $P_B$  and a data/chip select port (here designated as DATA). As described in such U. S. Patent, data may be written into, or read from, one of the memory array regions by control signals fed to either port  $P_A$  or to port  $P_B$ . In either case, the data fed to, or read from, the memory array region is on the common DATA port.

An exemplary one of the logic sections  $5010_1 - 5010_8$  will be discussed below in detail in connection with FIGS. 13A-15E and an exemplary one of the crossbar switches  $5004_1-5004_4$  in the logic networks  $221_{1B}-221_{2A}$  will be discussed below in detail in connection with FIGS. 10-12D. Suffice it to say here, however, each one of the memory array regions  $R_1-R_4$  is coupled to a pair of the logic sections  $5010_1, 5010_2; 5010_3, 5010_4; 5010_5, 5010_6; 5010_7, 5010_8$ , respectively, as shown. More particularly, each one of the logic sections  $5010_1, 5010_2; 5010_3, 5010_4; 5010_5, 5010_6; 5010_7, 5010_8$  includes: a pair of upper ports, Port\_A (A), Port\_B (B); a control port, C; and a data port, D, as indicated. The control port C of one each one of the logic sections  $5010_1, 5010_3, 5010_5, 5010_7$ , is coupled to port  $P_A$  of a corresponding one of the four memory array regions  $R_1-R_4$ . In like manner, the control port C of one of each one of the logic sections  $5010_2, 5010_4, 5010_6, 5010_8$  is coupled to port  $P_B$  of a corresponding one of the four memory array regions  $R_1-R_4$ , respectively as shown. Thus, each one of the memory array regions  $R_1-R_4$  is coupled to a redundant pair of the logic sections  $5010_1, 5010_2; 5010_3, 5010_4; 5010_5, 5010_6; 5010_7, 5010_8$ , respectively. The data ports D of logic section pairs  $5010_1, 5010_2; 5010_3, 5010_4; 5010_5, 5010_6; 5010_7, 5010_8$ , respectively, are coupled together and to the DATA port of a corresponding one of the memory regions,  $R_1-R_4$ , respectively, as indicated.

**Change to the Paragraphs beginning on Page 23 at line 19.**

As noted above in connection with FIG. 2, each one of the host computer processors  $121_1-121_{32}$  is coupled to here a pair (but not limited to a pair) of the front-end directors  $180_1-180_{32}$ , to provide redundancy in the event of a failure in one of the front end-directors  $181_1-181_{32}$  coupled thereto. Likewise, the bank of disk drives 140 has a plurality of, here 32, disk drives  $141_1-141_{32}$ , each disk drive  $141_1-141_{32}$  is coupled to here a pair (but not limited to a pair) of the back-end directors  $200_1-200_{32}$ , to provide redundancy in the event of a failure in one of the back-end

directors 200<sub>1</sub>-200<sub>32</sub> coupled thereto. Thus, the system has redundant front-end processor pairs 121<sub>1</sub>, 121<sub>2</sub> through 121<sub>31</sub>, 121<sub>32</sub> and redundant back-end processor pairs 141<sub>1</sub>, 141<sub>2</sub> through 141<sub>31</sub>, 141<sub>32</sub>. Considering the exemplary logic network 220<sub>1</sub> shown in FIGS. 9A, 9B and 9C, as noted above in connection with FIG. 8B, redundant front-end processor pairs 121<sub>1</sub> and 121<sub>2</sub>, are able to be coupled to ports M<sub>A1</sub> and M<sub>B1</sub> of a cache memory board. Thus, the ports M<sub>A1</sub> and M<sub>B1</sub> may be considered as redundant memory board ports. In like manner, the following may be considered as redundant memory ports because they are able to be coupled to a pair of redundant processors: M<sub>A2</sub> and M<sub>B2</sub>; M<sub>A3</sub> and M<sub>B3</sub>; M<sub>A4</sub> and M<sub>B4</sub>; M<sub>A5</sub> and M<sub>B5</sub>; M<sub>A6</sub> and M<sub>B6</sub>; M<sub>A7</sub> and M<sub>B7</sub>; and, M<sub>A8</sub> and M<sub>B8</sub>. It is noted that ports M<sub>A1</sub> and M<sub>B1</sub>; M<sub>A2</sub> and M<sub>B2</sub>; M<sub>A3</sub> and M<sub>B3</sub>; M<sub>A4</sub> and M<sub>B4</sub> are coupled to the front-end processors through front-end directors and ports M<sub>A5</sub> and M<sub>B5</sub>; M<sub>A6</sub> and M<sub>B6</sub>; M<sub>A7</sub> and M<sub>B7</sub>; M<sub>A8</sub> and M<sub>B8</sub> are coupled to the disk drives through back-end directors.

Referring again to FIGS. 9A, 9B and 9C, from the above it should be noted then that logic networks 221<sub>1B</sub> and 221<sub>1A</sub> may be considered as a pair of redundant logic networks (i.e., pair 1) because they are able to be coupled to redundant pairs of processors, here front-end processors. Likewise, logic networks 221<sub>2B</sub> and 221<sub>2A</sub> may be considered as a pair of redundant logic networks (i.e., pair 2) because they are able to be coupled to redundant pairs of disk drives. Further, logic network 221<sub>1B</sub> of pair 1 is coupled to upper port A of logic sections 5010<sub>1</sub>, 5010<sub>3</sub>, 5010<sub>5</sub>, and 5010<sub>7</sub> while logic network 221<sub>1A</sub> of pair 1 is coupled to port A of the logic sections 5010<sub>2</sub>, 5010<sub>4</sub>, 5010<sub>6</sub>, and 5010<sub>8</sub>. Logic network 221<sub>2B</sub> of pair 2 is coupled to port B of logic sections 5010<sub>1</sub>, 5010<sub>3</sub>, 5010<sub>5</sub>, and 5010<sub>7</sub> while logic network 221<sub>2A</sub> of pair 2 is coupled to port B of the logic sections 5010<sub>2</sub>, 5010<sub>4</sub>, 5010<sub>6</sub>, and 5010<sub>8</sub>.

**Change to Paragraphs beginning on Page 24 at line 26.**

Thus, the memory board 220<sub>1</sub> (FIGS. 9A, 9B and 9C) is arranged with a pair of independent fault domains: One fault domain, Fault Domain A, is associated with logic networks 221<sub>1B</sub> and 221<sub>2B</sub>, logic sections 5010<sub>1</sub>, 5010<sub>3</sub>, 5010<sub>5</sub>, 5010<sub>7</sub>, and ports P<sub>A</sub> of the memory array regions R<sub>1</sub>-R<sub>4</sub> and, the other fault domain, Fault Domain B, is associated with logic networks 221<sub>1A</sub> and 221<sub>2A</sub>, logic sections 5010<sub>2</sub>, 5010<sub>4</sub>, 5010<sub>6</sub>, 5010<sub>8</sub> and port P<sub>B</sub> of the memory array

regions  $R_1$ - $R_4$ . The logic in each one of the fault domains is operated by a corresponding one of a pair of independent clocks, Clock 1 and Clock 2 (FIGS. 9A, 9B and 9C). More generally, a fault domain is defined as a collection of devices which share one or more common points of failure. Here, Fault Domain A includes: logic networks  $221_{1B}$ ,  $221_{2B}$  (i.e., the S/Ps and crossbar switches  $5004_1$ - $5004_2$  therein) and logic sections  $5010_1$ ,  $5010_3$ ,  $5010_5$ ,  $5010_7$ , such devices being indicated by lines which slope from lower left to upper right (i.e., //). The other fault domain, Fault Domain B, includes: logic networks  $221_{1A}$ ,  $221_{2A}$  (i.e., the S/Ps and crossbar switches  $5004_3$ - $5004_4$  therein) and logic sections  $5010_2$ ,  $5010_4$ ,  $5010_6$ ,  $5010_8$ , such devices being indicated by lines which slope from upper left to lower right (i.e., \\\). It is noted from FIGS. 9A, 9B and 9C that port  $P_A$  of each one of the memory array regions  $R_1$ - $R_4$  is coupled to Fault Domain A while port  $P_B$  is coupled to fault domain B. Thus, each one of the fault domains includes the devices used to couple one of a pair of redundant processors to one of a pair of redundant control ports  $P_A$ ,  $P_B$  of the memory array regions  $R_1$ - $R_4$  and the other fault domain includes the devices used to couple the other one of the pair of redundant processors to the other one of a pair of redundant control ports  $P_A$ ,  $P_B$  of the memory array regions  $R_1$ - $R_4$ . As noted above each fault domain operates with a clock (i.e., clock 1, clock 2) separate from and independent of the clock used to operate the other fault domain.

Referring now to FIG. 10, an exemplary one of the crossbar switches  $5004_1$ - $5004_4$ , here crossbar switch  $5004_1$  is shown in detail to include four upper port interface sections A-D and lower port interface sections W-Z. The details of an exemplary one of the upper port interface sections A-D, here upper port interface section A, will be described in more detail in connection with FIGS. 11A, 11B, 11C and 11D and the details of an exemplary one of the lower port interface sections W-Z, here lower port interface section W, will be described in more detail in connection with FIGS. 12A, 12B, 12C and 12D. The function of the exemplary crossbar switch  $5004_1$  is to mediate the information cycle at the request of an initiating one of the directors coupled to one of the upper  $5006_1$ - $5006_4$  and one logic section  $5010_1$ - $5010_8$  indicated by the "tag" portion of the information (FIG. 16).

**Change to the Paragraph beginning on Page 26 at line 9.**

The lower port interface sections W-Z provides address, control, DATA and routing to one of the four of the logic sections 5010<sub>1</sub>-5010<sub>8</sub> (FIGS. 9A, 9B and 9C) in a manner to be described. Each one of the lower interface sections W-Z is adapted to couple a corresponding one of the four memory array regions R<sub>1</sub>-R<sub>4</sub> (FIGS. 9A, 9B and 9C), respectively, via logic sections 5010<sub>1</sub>-5010<sub>8</sub>. Each one of the four lower interface sections W-Z independently acts as an arbiter between the four upper interface sections A-D and the logic section 5010<sub>1</sub>-5010<sub>8</sub> coupled thereto. This allows for simultaneous transfers (i.e., information cycles) to multiple memory array regions R<sub>1</sub>-R<sub>4</sub> from multiple upper interface sections A-D. The upper interface section A-D are single threaded, i.e., one information cycle must be complete before another information cycle is allowed to the same memory array regions R<sub>1</sub>-R<sub>4</sub>.

**Change to the Paragraph beginning on Page 27 at line 1.**

More particularly, assume for example that information at upper port 5006<sub>4</sub> (FIGS. 9A, 9B and 9C) of crossbar switch 5004<sub>4</sub> is to be transferred to memory array region R<sub>1</sub>. Referring to FIG. 10 a negotiation, i.e., arbitration, must be made by lower port interface W as a result of a request made by the upper port interface section D of crossbar switch 5004<sub>4</sub> to section interface W thereof. When interface section W is available to satisfy such request, (i.e., not satisfying request from other one of the upper port interface sections A-C) interface W issues a grant to upper interface section D.

**Change to the Paragraphs beginning on Page 30 at line 19.**

As noted above, an exemplary one of the upper port interface sections A-D and an exemplary one of the lower port interface sections W-Z will be described in more detail in connection with FIGS. 11A-11D and 12A-12D, respectively. Suffice it to say here, however, that information fed to port 5006<sub>1</sub> is coupled to ports 5008<sub>1</sub>-5008<sub>4</sub> selectively in accordance with a "tag" portion such information. In a reciprocal manner, information fed to port 5008<sub>1</sub> is coupled

to ports 5006<sub>1</sub>-5006<sub>4</sub> selectively in accordance with the "tag" portion in such information. Further, ports 5006<sub>2</sub>-5006<sub>4</sub> operate in like manner to port 5006<sub>1</sub>, so that information at such ports 5006<sub>2</sub>-5006<sub>4</sub> may be coupled to ports 5008<sub>1</sub>-5008<sub>4</sub>. Still further, ports 5008<sub>2</sub>-5008<sub>4</sub> operate in like manner to port 5008<sub>1</sub>, so that information at such ports 5008<sub>2</sub>-5008<sub>4</sub> may be coupled to ports 5006<sub>1</sub>-5006<sub>4</sub>. It should also be noted that information may appear simultaneously at ports 5008<sub>1</sub> - 5008<sub>4</sub> with the information at one of such ports being coupled simultaneously to one of the ports 5006<sub>1</sub>-5006<sub>4</sub> while information at another one of the ports 5008<sub>1</sub> - 5008<sub>4</sub> is coupled to a different one of the ports 5006<sub>1</sub>-5006<sub>4</sub>. It is also noted that, in a reciprocal manner, information may appear simultaneously at ports 5006<sub>1</sub> - 5006<sub>4</sub> with the information at one of such ports being coupled simultaneously to one of the ports 5008<sub>1</sub>-5008<sub>4</sub> and with information at another one of the ports 5006<sub>1</sub> - 5006<sub>4</sub> being coupled to a different one of the ports 5008<sub>1</sub>-5008<sub>4</sub>.

Referring now to FIGS. 11A, 11B, 11C and 11D, an exemplary one of the upper port interface sections A-D, here upper port interface section A is shown in more detail. It is first noted that the information at port 5006<sub>1</sub> includes: the "tag" portion referred to above; an address CRC ADDR\_CRC portion, an address ADDR portion, a memory control portion (i.e., read/write, transfer length, "Wait and Validate", etc.); a data portion, (DATA); and a DATA Cyclic Redundancy Check (CRC) portion (DATA\_CRC).

**Change to the Paragraph beginning on Page 33 at line 1.**

It is noted that the upper port section A also includes a memory board checker 5114. Each of the here eight memory board 220<sub>1</sub>-220<sub>8</sub> (FIG. 8) plugs into the backplane 302 as discussed above in connection with FIG. 3. As noted above, here the backplane 302 is adapted to a plurality of, here up to eight memory boards. Thus, here the backplane 302 has eight memory board slots. Pins P<sub>1</sub>-P<sub>3</sub> (FIGS. 9A, 9B and 9C) are provided for each backplane 320 memory board slot and produce logic voltage levels indicating the slot position in the backplane. Thus, here the slot position may be indicated with the logic signals on the three pins P<sub>1</sub>-P<sub>3</sub> to produce a three bit logic signal representative of the backplane slot position. Referring again to FIGS. 9A, 9B and 9C, the exemplary memory board 220<sub>1</sub> is shown plugged into a slot in the backplane 302.



As noted above, the slot has pins  $P_1$ - $P_3$  which provides the slot position three bit logic signal indicative of the slot or "memory board" number in the backplane. The logic signals produced by the pins  $P_1$ - $P_3$  are fed to the memory board checker 5114 (FIGS. 11A, 11B, 11C and 11D). Also fed to the memory board checker 5114 are the 3-bits of the "tag" which indicates the one of the memory array boards which is to receive the data (i.e., a 3-bit "memory board code"). If the three bit memory board indication provided by "tag" is the same as the backplane slot or "memory board number" indication provided by the pins  $P_1$ - $P_3$ , the director routed the information cycle to the proper one of the eight memory boards and such "accept" indication is provided to the decode logic/ADDR CRC checker 5112 via line A/R. On the other hand, if the three bit memory board indication provided by "tag" is different from the backplane slot indication provided by the pins  $P_1$ - $P_3$ , the information cycle was not received by the correct one of the memory boards and such "reject" indication is provided to the decode logic/ADDR CRC checker 5112 via line A/R. When a reject indication is provided to the decode logic/ADDR CRC checker 5112, the intended transfer is prevented and the indication is provided by the decode logic/ADDR CRC checker 5112 to the initiating director via the A/R line. Thus, if the "memory board number" provided by pins  $P_1$ - $P_3$  does not match the "memory board code" contained in the "tag" the transfer request from the director is rejected and such error indication is sent back to the director. In this manner, a routing error in the director is detected immediately and is not propagated along.

**Change to the Paragraphs beginning on Page 34 at line 16.**

It is noted that the decode logic and ADDR CRC checker 5112 in upper port interface logic A also produces request signals RWA, RXA, RYA, and RZA and sends such request signal to lower port sections W-Z, respectively. Such requests are fed to an arbitration logic 5114 (FIGS. 12A, 12B, 12C and 12D) included within each of the lower port sections W, X, Y and Z, respectively. Thus, because the other upper port sections B-D operate in like manner to upper port section A, the arbitration 5114 in lower port interface section W may receive requests RWB, RWC, and RWD from such other upper port sections B-D, respectively. In accordance with a predetermined arbitration rule, such as, for example, first-come, first-served, the arbitration logic

5114 of lower port interface section W grants for access to lower port 5008<sub>1</sub> of lower port section W to one of the requesting upper port sections A-D via a grant signal on one of the lines GWA, GWB, GWC and GWD, respectively.

Thus, referring again to FIGS. 11A, 11B, 11C and 11D, the decode logic/CRC ADR checker 5112 issues a request on line RWA when port 5008<sub>1</sub> (FIG. 10) desires, based on the two bit information in the "tag", memory array region R<sub>1</sub> (FIGS. 9A, 9B and 9C). In like manner, if memory array regions R<sub>2</sub>-R<sub>4</sub> are indicted by the "tag", requests are made by the upper port section on lines RXA, RYA, RZA, respectively. The other upper port sections B-D operate in like manner. The grants (GR) produced by the lower port sections W, X, Y and Z are fed to the upper port sections A-D as indicated above. Thus, considering exemplary upper port section A (FIGS. 11A, 11B, 11C and 11D), the grant signals from lower port sections W-Z are fed to the decode logic/CRC checker 5112 therein on lines GWA, GXA, GYA and GZA, respectively. When a grant on one of these four lines GWA, GXA, GYA and GZA is received by the decode logic/CRC checker 5112, such checker 5112 enables the gating signal to be produced on the one of the enable lines EAW, EAX, EAY, EAZ indicated by the "tag" portion. For example, if the "tag" indicates that memory array region R<sub>3</sub> (which is adapted for coupling to port 5008<sub>3</sub> of lower port section Y) the checker 5112 issues a request on line RYA. When after the arbitration logic 5114 in section Y determines that lower port logic A is to be granted access to port 5008<sub>3</sub>, such lower port section Y issues a grant signal on line GYA. In response to such grant, the checker 5112 issues an enable signal on line EAY to thereby enable information to pass to port A<sub>3</sub> (FIGS. 11A, 11B, 11C and 11D).

**Change to the Paragraph beginning on Page 35 at line 19.**

Referring now to FIGS. 12A, 12B, 12C and 12D, exemplary lower port section W is shown to include arbitration logic 5114 described above, and the selector 5120 fed by signals on ports W<sub>1</sub>-W<sub>4</sub>. (Referring again to FIG. 10, ports W<sub>1</sub>-W<sub>4</sub> are coupled to ports A<sub>1</sub>, B<sub>1</sub>, C<sub>1</sub> and D<sub>1</sub>, respectively, of upper port interface sections A-D, respectively.) Thus, when the arbitration logic 5114 grants access to one of the upper port sections A-D, the decoder 5122 decodes the grant

information produced by the arbitration logic and produces a two bit control signal for the selector 5120. In response to the two bit control signal produced by the decoder 5122, the selector couples one of the ports  $W_1$ - $W_4$  (and hence one of the upper port sections A-D, respectively), to the output of the selector 5120 and hence to lower port 5008<sub>1</sub> in a manner to be described.

**Change to the Paragraphs beginning on Page 36 at line 22.**

More particularly, the DATA, memory control, ADDR, and "tag" portions (with their byte parity (p) generated by parity generator 5102 (FIGS. 11A, 11B, 11C and 11D)) of the information coupled to the output of selector 5120 is stored in the register 5124. As noted above in connection with FIG. 16, the DATA\_CRC portion (i.e., the words X and Y) occurs after the last DATA word.

Thus, as the words in the DATA clock through register 5124 they pass into the DATA\_CRC checker 5132 where the CRC of the DATA is determined (i.e., the DATA\_CRC checker 5132 determine X and Y words of the DATA fed to such checker 5132). The actual X and Y words (i.e., DATA\_CRC stored in register 5128, both content (n) and parity (p)) are stored successively in register 5128 and are then passed to checker 5132 where they are checked against the X and Y words determined by the checker 5132. As noted above, the DATA has appended to it its parity (p). Thus, the "information" whether in register 5124 or register 5128 has a content portion indicated by "n" and its parity indicated by "p". Thus, the DATA\_CRC register 5128 includes the DATA\_CRC previously stored in register 5104<sub>1</sub> (FIGS. 11A, 11B, 11C and 11D) (i.e., the content portion designated by "n") and its parity (designated by "p"). The DATA, memory control, ADDR, and "tag" portions, (with their parity (p) (i.e., content "n" plus its appended parity "p") stored in register 5124 may be coupled through a selector 5149 through one of two paths: One path is a direct path when the "Wait and Validate" command is not issued by the director; and, a second path which includes a delay network 5130, here a three clock pulse delay network 5130.

More particularly, it is noted that the DATA, control, ADDR, "tag", both content (n) and

parity (p) are also fed to a DATA\_CRC checker 5132. Also fed to the DATA\_CRC checker 5132 is the output of DATA\_CRC register 5128. The CRC checker 5132 checks whether the DATA\_CRC (content "n" plus its parity "p") is the same as the CRC of the DATA, such DATA having been previously stored in register 5104<sub>2</sub> (FIGS. 11A, 11B, 11C and 11D), i.e., the content "n" plus its parity "p" of the DATA previously stored in register 5104<sub>2</sub> (FIGS. 11A, 11B, 11C and 11D). If they are the same, (i.e., no DATA\_CRC\_ERROR), a logic 0 is produced by the CRC checker 5132. If, on the other hand, they are not the same, (i.e., a DATA\_CRC\_ERROR), the CRC checker 5132 produces a logic 1. The output of the Data\_CRC checker 5132 thereby indicates whether there is an error in the CRC of the DATA. Note that a DATA\_CRC\_ERROR is not known until three clock cycles after the last sixteen-bit portion of the DATA (i.e., the word of the DATA, FIG. 16) is calculated due to the nature of the CRC algorithm. Such indication is fed to a selector 5152 via an OR gate 5141. If there is a DATA\_CRC\_ERROR, the "information" at the output of the delay network 5130 (i.e., the last word of the DATA (FIG. 16)) with its parity (p)) is corrupted. Here, the content (n) of such "information" (i.e., the "information" at the output of the delay network 5130 (i.e., the last word of the DATA (FIG. 16))) is fed to a second input I<sub>2</sub> of the selector 5140. The parity (p) of such "information" (i.e., the last word of the DATA (FIG. 16)) is fed non-inverted to one input of selector 5152 and inverted, via inverter 5150, to a second input of the selector 5152. If there is a DATA\_CRC\_ERROR detected by data CRC checker 5132, the inverted parity is passed through the selector 5152 and appended to the content portion (n) of the "information" (i.e., the last word of the DATA (FIG. 16)) provided at the output of the delay network 5130 and both "n" and appended "p" are fed to the second input I<sub>2</sub> of selector 5140 thereby corrupting such "information". It should be noted that the remaining portions of the information cycle (i.e., the memory control, address (ADDR), "tag", and all but the last word of the DATA (FIG. 16)) pass through the delay network 5130 without having their parity (p) corrupted.

If there is a no "Wait and Validate" transfer, logic decoder 5122 selects the first input I<sub>1</sub> as the output of the selector 5140. If there is a "Wait and Validate" transfer, the logic decoder 5122 selects the second input I<sub>2</sub> as the output of the selector 5140. It is noted, however, that that

because the last word of DATA (FIG. 16) is delayed three clock pulses (from Clock 1) by registers 5142, 5144, and 5146 (such registers 5142, 5144 and 5146 being fed by such Clock 1), the DATA\_CRC check is performed before the last word of the DATA appears at the output of register 5146. Thus, the last word of the DATA is corrupted in byte parity before being passed to the logic section 5010<sub>1</sub>-5010<sub>8</sub>. That is, because of the delay network 5130, the DATA\_CRC is evaluated before the last word of the DATA has passed to port 5008<sub>1</sub>. This corruption in parity (p), as a result of a detected DATA\_CRC error, is detected by a parity checker 6106 (FIGS. 14A, 14B, 14C and 14D) in the following logic section 5010<sub>1</sub>-5010<sub>8</sub> in a manner to be described. Suffice it to say here, however, that detection of the parity error (produced by the detected CRC error) prevents such corrupted information from storage in the SDRAMs.

On the other hand, if there is no DATA\_CRC\_ERROR (and no error in the parity of the DATA\_CRC detected by the parity checker 6106 (FIGS. 14A, 14B, 14C and 14D) in a manner to be described) the non-inverted parity (p) is appended to the "information" (i.e., DATA, memory control, ADDR, and "tag") provided at the output of the delay network 5130 and such information is fed to the proper memory address region R<sub>1</sub>-R<sub>4</sub> as indicated by "tag".

**Change to the Paragraphs beginning on Page 39 at line 28.**

Thus, the exemplary lower port interface section W (FIGS. 12A, 12B, 12C and 12D) includes a parity generator made up of an exclusive OR gate 5134 and register 5136 arranged as shown fed by the parity (p) of the DATA portion stored in register 5124. The generated parity p is fed to a comparator 5138 along with the parity (p) of the DATA\_CRC (i.e., DATA\_CRC\_PARITY), as indicated. If the two are the same at the end of the DATA portion of the information cycle (FIG. 16), a logic 0 is produced by the comparator 5138 and such logic 0 passes to the selector 5152 to enable the non-inverted parity to pass through such selector 5152. If there is an error in the parity bit of the CRC, a logic 1 is produced by the comparator 5138 and the inverted parity is passed through the selector 5152. The logic 1 output of comparator 5138 passes through OR gate 5141 to couple the inverted parity (p) through selector 5152 to append to the content port (n) of DATA control, ADDR, and "tag" at port I<sub>2</sub> of selector 5140. Thus, if there

is either a DATA\_CRC\_ERROR or if DATA\_CRC\_PARITY is different from parity of the DATA\_PARITY at the end of the DATA portion of the information cycle as indicated by a signal produced on line COMP\_ENABLE by the logic decoder 5122, a logic 1 is produced at the output of OR gate 5141 thereby coupling the inverted parity through selector 5152. Otherwise, the non-inverted parity passes through selector 5152. That is, the COMP\_EN is produced at the end of the DATA in the information cycle (FIG. 16).

It is noted that information read from the memory region passes to a register 5170 and a CRC generator 5172. The generated CRC is appended to the information clocked out of the register 5170. Four copies of the information with appended CRC are stored in registers 5174<sub>1</sub>-5174<sub>4</sub>, respectively. In response to the "tag" portion fed to logic decoder 5122, a selected one of the registers 5174<sub>1</sub>-5174<sub>4</sub> is coupled to one of the port W<sub>1</sub>-W<sub>4</sub> by selector 5180 and gates 5182<sub>1</sub>-5182<sub>4</sub> in a manner similar to that described in connection with FIGS. 11A, 11B, 11C and 11D.

Referring now to FIGS. 13A, 13B, 13C, 13D and 13E, a pair of the logic sections 5010<sub>1</sub>-5010<sub>8</sub> (memory array region controllers), here logic sections 5010<sub>1</sub> and 5010<sub>2</sub> are shown. As noted above in connection with FIGS. 9A, 9B and 9C, both logic sections 5010<sub>1</sub> and 5010<sub>2</sub> are coupled to the same memory array region, here memory array region R<sub>1</sub>. As was also noted above in connection with FIGS. 9A, 9B and 9C, the logic section 5010<sub>1</sub> is in one fault domain, here fault domain A, and logic section 5010<sub>2</sub> is in a different fault domain, here fault domain B. Thus, logic section 5010<sub>1</sub> operates in response to clock pulses from Clock 1 and logic section 5010<sub>2</sub> operates in response to clock pulses from Clock 2.

As noted above, each logic section 5010<sub>1</sub>-5010<sub>8</sub> (FIGS. 9A, 9B and 9C) includes a pair of upper ports, A and B, a control port C and a data port D. Referring to FIGS. 13A, 13B, 13C, 13D and 13E, an exemplary logic section 5010<sub>1</sub> is shown in detail to include a upper port A controller 6002A coupled to upper port A, a upper port B controller 6002B coupled to upper port B, and a memory refresh section 6002R.

**Change to the Paragraph beginning on Page 41 at line 21.**

Here, the arbitration algorithm is an asymmetric round robin sharing of the common

memory array region  $R_1$ . The arbitration logic 6004<sub>1</sub>, 6004<sub>2</sub> and the algorithm executed therein will be described in more detail in connection with FIGS. 15A, 15B, 15C, 15D and 15E. Suffice it to say here however that the arbitration grants access to the common memory array region based on the following conditions:

**Change to the Paragraphs beginning on Page 44 at line 3.**

Referring again to FIGS. 13A, 13B, 13C, 13D and 13E, the arbitration logic 6004<sub>1</sub>, 6004<sub>2</sub> in each one of the logic sections 5010<sub>1</sub>, 5010<sub>2</sub> produces: a memory output enable (MOE) signal; a memory refresh enable (MRE) signal (to be described in more detail in connection with FIGS. 15A-15E and 19); and, a memory grant (MG) signal, (to be described in more detail in connection with FIGS. 15A-15E and 19). Thus, logic section 5010<sub>1</sub> produces a memory output enable signal MOEA (to be described in more detail in connection with FIGS. 15A-15E and 19), a memory refresh enable signal MREA (to be described in more detail in connection with FIGS. 15A-15E and 19) and a memory grant signal MGA (to be described in more detail in connection with FIGS. 15A-15E and 19). Likewise, logic section 5010<sub>2</sub> produces a memory output enable signal MOEB (to be described in more detail in connection with FIGS. 15A-15E and 19), a memory refresh enable signal MREB (to be described in more detail in connection with FIGS. 15A-15E and 19) and a memory grant signal MGB (to be described in more detail in connection with FIGS. 15A-15E and 19). Suffice it to say here, however, that the MOEA signal is a triplicate signal MOE<sub>I-1</sub>, MOE<sub>I-2</sub>, MOE<sub>I-3</sub> and the MGA signal is also a triplicate signal MGE<sub>IA</sub>, MGE<sub>IIA</sub>, and MGE<sub>IIIA</sub>.

The MOEA and MREA signals from the logic section 5010<sub>1</sub> and the MOEB and MREB signals from the logic section 5010<sub>2</sub> are fed to a watch dog (WD) section 6006, to be described in more detail in connection with FIGS. 15A, 15B, 15C, 15D and 15E. Suffice it to say here, however, that, as noted above, the arbitration algorithm is a function of the operating/non-operating condition of the logic sections 5010<sub>1</sub>, 5010<sub>2</sub>. This operating/non-operating condition is determined by the watchdog section 6006 and more particularly by examining the MOEA, MREA, MOEB, MREB signals produced by the logic sections 5010<sub>1</sub> and 5010<sub>2</sub>, 6002B,

respectively. The MOEA, MREA, MOEB, MREB signals are asserted when there is a grant. Such signals MOEA, MREA, MOEB, MREB are fed to the watchdog section 6006. As will be described, the watchdog section 6006 examines the time history of these signals to determine if the logic section 5010<sub>1</sub> or 5010<sub>2</sub> asserting them is operating properly. Based on the results of such examination, the watchdog selects the Condition I, Condition II, or Condition III, described above.

More particularly, consider, for example, a case where the MOEA signal is asserted for too long a predetermined time interval. It should be recalled that the logic section 5010<sub>1</sub> producing such MOEA signal is granted access to the memory in State 1 of the normal arbitration condition (i.e., Condition I, above). The watchdog section 6006 thus detects a fault in logic section 5010<sub>1</sub>. When such a fault is detected, the watchdog section 6006 issues a Condition III signal on in triplicate on lines MSAB to the arbitration sections 6004<sub>1</sub>, 6004<sub>2</sub> in both the logic sections 5010<sub>1</sub>, 5010<sub>2</sub>, respectively, indicating that the arbitration algorithm will operate in accordance with the States set forth above for Condition III. Further, the watchdog 6006 issues a data output enable signal in triplicate on lines DOEA (i.e., DOEA<sub>0</sub>, DOEA<sub>1</sub>, and DOEA<sub>2</sub>). This triplicate signal DOEA (i.e., DOEA<sub>0</sub>, DOEA<sub>1</sub>, and DOEA<sub>2</sub>) is fed to a majority gate (MG) 6007 (FIGS. 13A, 13B, 13C, 13D and 13E), in accordance with the majority of the triplicate data fed to it, provides an enable/disable signal for gate 6009. If the majority indicates a fault, the gate 6009 inhibits DATA from passing between the logic section 5010<sub>1</sub> and the data port D thereof.

Consider the case where the arbitration is in Condition I. Consider also that in such condition I, the MREA signal is not produced after a predetermined time interval which ensures proper refreshing on the SDRAMs in the memory array region R<sub>1</sub>. The watchdog section 6006 will again detect a fault in the logic section 5010<sub>1</sub> port A controller 6002A. When such a fault is detected, the watchdog section 6006 issues a Condition III signal on in triplicate on lines MSAB (i.e., MSAB<sub>0</sub>, MSAB<sub>1</sub>, MSAB<sub>2</sub>) to the arbitration sections 6004<sub>1</sub>, 6004<sub>2</sub> in both the logic sections 5010<sub>1</sub>, 5010<sub>2</sub>, respectively. Further, the watchdog 6006 issues a data output enable signal in triplicate on lines DOEA (i.e., DOEA<sub>0</sub>, DOEA<sub>1</sub>, and DOEA<sub>2</sub>) (FIGS. 13A, 13B, 13C, 13D and 13E) to inhibit DATA from passing between the logic section 5010<sub>1</sub> and the data port D thereof.



**Change to the Paragraphs beginning on Page 46 at line 11.**

Thus, referring to FIGS. 13A, 13B, 13C, 13D and 13E, the memory array region  $R_1$  may be coupled to either Port\_A (A) or Port\_B (B) of the logic sections 5010<sub>1</sub>, 5010<sub>2</sub> or to the memory refresh section 6002R therein selectively in accordance with a Port\_A\_SELECT, Port\_B\_SELECT, Port\_R\_SELECT signal fed to a pair of selectors 6010<sub>C</sub>, 6010<sub>D</sub>, shown in more detail for exemplary logic section 5010<sub>1</sub>. Access by the upper port A controller 6002A (i.e., Port\_A), by the upper port B controller 6002B, or the memory refresh section 6002R to the memory array region  $R_1$  is in accordance with the algorithm described above

An exemplary one of the upper port A and port B logic controllers 6002A and 6002B, here controller 6002A, will be described in more detail in connection with FIGS. 14A, 14B, 14C and 14D. Suffice it to say here, however, that it is noted that the output of selector 6010<sub>C</sub> is coupled to the control port C of the exemplary logic section 5101<sub>1</sub> and the output of selector 6010<sub>D</sub> is coupled to the data port D of the exemplary logic section 5101<sub>1</sub> through the gate 6009. Each one of the selectors 6010<sub>C</sub> and 6010<sub>D</sub> has three inputs A, B, and R, as shown. The A, B and R inputs of selector 6010<sub>C</sub> are coupled to: the ADR/CONTROL produced at the output of upper port A controller 6002A; the ADR/CONTROL produced at the output of upper port B controller 6002B; and, the portion REFRESH\_C of the refresh signal produced by the memory refresh section 6002R, respectively as indicated. The A, B and R inputs of selector 6010<sub>D</sub> are coupled to: the WRITE DATA produced at the output of upper port A controller 6002A; the WRITE DATA produced at the output of upper port B controller 6002B; and, the portion REFRESH\_D of the refresh signal produced by the memory refresh section 6002R, respectively as indicated. The Port\_A\_SELECT, Port\_B\_SELECT are produced by the upper port A controller 6002A, upper port B controller 6002B in a manner to be described. The Port\_R\_SELECT signal is produced by the memory refresh section 6002R in a manner to be described to enable proper operation of the above described arbitration algorithm and to proper a refresh signal to the SDRAMs in the memory array region  $R_1$  at the proper time. Suffice it to say here, however, that when port A controller 6002A produces the Port\_A\_SELECT signal, the ADR/CONTROL at the

output of port A controller 6002A passes to the output of the selector 6010C and the DATA\_WRITE at the output of the port A controller 6002A passes to the output of the selector 6010D. Likewise, when port B controller 6002B produces the Port\_B\_SELECT signal, the ADR/CONTROL at the output of port B controller 6002B passes to the output of the selector 6010C and the DATA\_WRITE at the output of the port B controller 6002B passes to the output of the selector 6010D. In like manner, when refresh memory section 6002R produces the Port\_R\_SELECT\_C signal, the REFRESH\_C at the output of refresh memory section 8002R passes to the output of the selector 6010C and in response to the Port\_R\_SELECT signal, the REFRESH\_D at the output of the refresh memory section 8002R passes to the output of the selector 6010D.

**Change to the Paragraphs beginning on Page 47 at line 19.**

Referring now to FIGS. 14A, 14B, 14C and 14D, the exemplary port A controller 6002A is shown in more detail to include a Port A primary control section 6100P and a Port A secondary control section 6100S. The two sections 6100P and 6100S are both coupled to port A and both implement the identical control logic. Thus, each one of the two sections 6100P and 6100S should produce the same results unless there is an error, here a hardware fault, in one of the two sections 6100P and 6100S. Such a fault is detected by a fault detector 6102 in a manner to be described.

Thus, referring to the details of one of the two sections 6100P and 6100S, here section 6100P, it is first noted that the information at Port\_A is fed to a parity checker 6101. It is noted that there is an error in parity induced by the CRC check described in FIGS. 12A, 12B, 12C and 12D in connection with selector 5152, such detected parity error is reported to a control and DATA path logic 6112. In response to a detected parity error, control and DATA path logic 6112 prevents memory control signals (e.g., suppress the Column Address Select signal to the SDRAMs) from being produced on the CONTROL\_P line. Thus, absent control signal, DATA will not be stored in the memory region.

**Change to the Paragraph beginning on Page 50 at line 18.**

It is noted that the RAP and PAS signals are both sent to the arbitration logic 6004<sub>1</sub> (FIGS. 13A, 13B, 13C, 13D and 13E) as composite signal REQUEST A. The arbitration section 6004<sub>1</sub> considers a valid request only if both signals RAP and RAS are the same. In like manner, the arbitration logic 6004<sub>1</sub> issues separate grant signals GAP and GAS which are shown in FIGS. 13A, 13B, 13C, 13D and 13E as a composite signal GRANT\_A. Likewise, PORT\_A\_P\_SELECT and PORT\_A\_S\_SELECT signals are both sent to the arbitration logic 6004<sub>1</sub> (FIGS. 13A, 13B, 13C, 13D and 13E) as composite signal PORT\_A\_SELECT. The arbitration section 6004<sub>1</sub> considers a valid request only if both signals PORT\_A\_P\_SELECT and PORTA\_S\_SELECT are the same.

**Change to the Paragraphs beginning on Page 51 at line 1.**

Referring now to FIGS. 15A, 15B, 15C, 15D and 15E, the arbitration logics 6004<sub>1</sub>, 6004<sub>2</sub> of the logic sections 5010<sub>1</sub>, 5010<sub>2</sub>, respectively, are shown along with the watchdog section 6006. It is first noted that the arbitration logic 6004<sub>1</sub>, 6004<sub>2</sub> are identical in construction.

Arbitration logic 6004<sub>1</sub> is fed by:

REQUEST A (i.e., RAP, RAS) from upper port A controller 6002A of logic section 5010<sub>1</sub> (FIGS. 13A, 13B, 13C, 13D and 13E);

REQUEST B (RBP, RBS) from upper port B controller 6002B of logic section 5010<sub>1</sub> (FIGS. 13A, 13B, 13C, 13D and 13E);

REQUEST R from upper memory refresh section 6002R of logic section 5010<sub>1</sub> (FIGS. 13A, 13B, 13C, 13D and 13E) (It is to be noted that the REQUEST R is made up of two signals, each being produced by identical primary and secondary identical memory refresh units, not shown, in memory refresh section 6002R both of which have to produce the same refresh signal in order for the arbitration logic 6004<sub>1</sub> to respond to the refresh request).

Arbitration logic 6004<sub>2</sub> is fed by:

REQUEST A from upper port A controller 6002A of logic section 5010<sub>2</sub> (FIGS. 13A, 13B, 13C, 13D and 13E);

REQUEST B from upper port B controller 6002B of logic section 5010<sub>2</sub> (FIGS. 13A, 13B, 13C, 13D and 13E);

REQUEST R from upper memory refresh section 6002R of logic section 5010<sub>2</sub>.

As shown in FIGS. 15A, 15B, 15C, 15D and 15E, each one of the three request signals REQUEST A, REQUEST B, and REQUEST R, produced in logic section 5010<sub>1</sub> is fed, in triplicate, to three identical arbitration units, (i.e., arbitration unit I, arbitration unit II, and arbitration unit III) in the arbitration logic 6004<sub>1</sub> of such logic section 5010<sub>1</sub>, as indicated. (See also FIG. 19). Likewise, each one of the three request signals REQUEST A, REQUEST B, and REQUEST R, produced in logic section 5010<sub>2</sub> is fed, in triplicate, to three identical arbitration units, (i.e., arbitration unit I, arbitration unit II, and arbitration unit III, in the arbitration logic 6004<sub>2</sub> of such logic section 5010<sub>2</sub> as indicated.

**Change to the Paragraph beginning on Page 53 at line 17.**

The signals MSAB<sub>I\_1</sub>, MSAB<sub>II\_1</sub>, and MSAB<sub>III\_1</sub>, are fed to the arbitration units I, II, and III, respectively, in arbitration logic 6004<sub>1</sub>. In like manner, the signals MSAB<sub>I\_2</sub>, MSAB<sub>II\_2</sub>, and MSAB<sub>III\_2</sub>, are fed to the arbitration units I, II, and III, respectively, in arbitration logic 6004<sub>2</sub>. In response to such signals, each one of the arbitration units I, II, and III, makes an independent determination of whether logic section 5010<sub>1</sub> (FIGS. 13A, 13B, 13C, 13D and 13E) or logic section 5010<sub>2</sub> will be granted access to the memory array region R<sub>1</sub>. A grant by logic section 5010<sub>1</sub> to logic section 5010<sub>2</sub> is indicated by a Memory Grant (MG) signal. Thus, arbitration units I, II, and III of logic section 5010<sub>1</sub> produce Memory Grant signals MGI<sub>1</sub>, MGII<sub>1</sub>, and MGIII<sub>1</sub>, respectively. Such signals are fed to a synchronization filter 6202<sub>2</sub> in arbitration logic 6004<sub>2</sub>. The synchronization filter 6202<sub>2</sub> operates as is constructed in the same manner as synchronization filters 6200<sub>1</sub> and 6200<sub>2</sub>. In like manner arbitration units I, II, and III of logic section 5010<sub>2</sub> produce Memory Grant signals MGI<sub>2</sub>, MGII<sub>2</sub>, and MGIII<sub>2</sub>, respectively. Such signals are fed to a synchronization filter 6202<sub>1</sub> in arbitration logic 6004<sub>1</sub>. The synchronization filter 6202<sub>1</sub> operates as is constructed in the same manner as synchronization filter 6202<sub>2</sub>.

**Change to the Paragraph beginning on Page 54 at line 25.**

Referring now to FIGS. 20A, 20B and 20C, the three arbitrations I, II, and III of exemplary arbitration logic 6004<sub>1</sub> are the signals fed thereto and produced thereby are shown in more detail. It is first noted that the primary signal REQUEST\_A\_P, (RAP), and the secondary request signal REQUEST\_A\_S (RAS) are each fed in triplicate; one copy to each of the arbitrations I, II, and III, as indicated. The one of the triplicate RAP and RAS fed to arbitration I are fed to an AND gate 8000<sub>1</sub>, a second one of the triplicate RAP and RAS fed to arbitration II are fed to an AND gate 8000<sub>2</sub>, and the third one of the triplicate RAP and RAS fed to arbitration III are fed to an AND gate 8000<sub>3</sub>, as indicated. Likewise, the signals REQUEST\_B\_P, (RBP), and REQUEST\_B\_S (RBS) are each fed in triplicate; one copy to each of the arbitrations I, II, and III, as indicated. The one of the triplicate RBP and RBS fed to arbitration I are fed to an AND gate 8002<sub>1</sub>, a second one of the triplicate RBP and RBS fed to arbitration II are fed to an AND gate 8002<sub>2</sub>, and the third one of the triplicate RBP and RBS fed to arbitration III are fed to an AND gate 8002<sub>3</sub>, as indicated. As mentioned briefly above, there are two memory refresh units in the memory refresh section 6002R (FIGS. 13A, 13B, 13C, 13D and 13E). One, a primary unit (not shown), issues a request RRP and the other, a secondary unit (not shown), issues a request RRS. Above, in connection with FIGS. 13A, 13B, 13C, 13D and 13E, these two requests were considered as a composite request (REFRESH\_REQUEST) to simplify the discussion presented above. Here, in connection with FIG. 19, the individual signals RRP, RRS are shown in more detail. Thus, the signals RRP, RRS are each fed in triplicate; one copy to each of the arbitrations I, II, and III, as indicated. The one of the triplicate RRP and RRS is fed to arbitration I are fed to an AND gate 8004<sub>1</sub>, a second one of the triplicate RRP and RRS fed to arbitration II are fed to an AND gate 8004<sub>2</sub>, and the third one of the triplicate RRP and RS fed to arbitration III are fed to an AND gate 8004<sub>3</sub>, as indicated.

**Change to the Paragraphs beginning on Page 55 at line 20.**

Each arbitration I, II and II issues pairs of grants, i.e., a primary grant to the primary unit and a secondary grant to the secondary unit. Thus, each of the arbitrations I, II and III issues: the

primary and secondary grants (GAP and GAS, respectively) to the Port A primary control section 6100P (FIGS. 14A, 14B, 14C and 14D) and Port A secondary control section 6100S of Port A controller 6002A; the primary and secondary grants (GBP and GBS, respectively) to the Port B primary control section and Port A secondary control section of Port B controller 6002B; and the primary and secondary grants (GRP and GRS, respectively) to the memory refresh primary unit memory refresh secondary unit of the memory refresh section 6002R (FIGS. 13A, 13B, 13C, 13D and 13E).

The arbitrations I, II, and III produce Memory Output Enable signals  $MOE_{I-1}$ ,  $MOE_{II-1}$ , and  $MOE_{III-1}$ , respectively, as indicated, for the watchdogs  $WD_I$ ,  $WD_{II}$  and  $WD_{III}$ , respectively, as shown in FIGS. 15A, 15B, 15C, 15D and 15E. The arbitrations I, II, and III produce Memory Refresh Enable signals  $MRE_{I-1}$ ,  $MRE_{II-1}$ , and  $MRE_{III-1}$ , respectively, as indicated, for the watchdogs  $WD_I$ ,  $WD_{II}$  and  $WD_{III}$ , respectively, as shown in FIGS. 15A, 15B, 15C, 15D and 15E. The arbitrations I, II, and III produce Memory Grant signals  $MG_I$ ,  $MG_{II}$ , and  $MG_{III}$ , respectively, as indicated, for the registers 6204<sub>I</sub>, 6204<sub>II</sub> and 6204<sub>III</sub>, respectively, of filter 6202<sub>2</sub> of logic section 5010<sub>2</sub>, as shown in FIG. 15.